```python
In [2]:  1  def probAttr(data,attr,val):
         2      Total = data.shape[0]
         3      cnt= len(data[data[attr]==val])
         4      return cnt,cnt/Total
         5
         6  def train(data,Attr,conceptVals,concept):
         7      conceptProbs={}
         8      countConcept={}
         9      for cVal in conceptVals:
        10          countConcept[cVal],conceptProbs[cVal]=probAttr(data,concept,cVal)
        11      AttrConcept={}
        12      probability_list={}
        13      for att in Attr:
        14          probability_list[att]={}
        15          AttrConcept[att]={}
        16          for val in Attr[att]:
        17              AttrConcept[att][val]={}
        18              a,probability_list[att][val]=probAttr(data,att,val)
        19              for cVal in conceptVals:
        20                  dataTemp=data[data[att]==val]
        21                  AttrConcept[att][val][cVal]=len(dataTemp[dataTemp[concept]==cVal])/countConcept[cVal]
        22      print(f"P(A):{conceptProbs}\n")
        23      print(f"P(X/A):{AttrConcept}\n")
        24      print(f"P(X):{probability_list}\n")
        25      return conceptProbs,AttrConcept,probability_list
        26
        27  def test(examples,Attr,concept_list,conceptProbs,AttrConcept,probability_list):
        28      misclassification_count=0
        29      Total=len(examples)
        30      for ex in examples:
        31          px={}
        32          for a in Attr:
```

```python
29    Total=len(examples)
30    for ex in examples:
31        px={}
32        for a in Attr:
33            for x in ex:
34                for c in concept_list:
35                    if x in AttrConcept[a]:
36                        if c not in px:
37                            px[c]=conceptProbs[c]*AttrConcept[a][x][c]/probability_list[a][x]
38                        else:
39                            px[c]=px[c]*AttrConcept[a][x][c]/probability_list[a][x]
40        print(px)
41        classification=max(px,key=px.get)
42        print(f"classfication:{classification} Excepted:{ex[-1]}")
43        if(classification!= ex[-1]):
44            misclassification_count += 1
45            misclassification_rate = misclassification_count * 100/Total
46            accuracy = 100 - misclassification_rate
47    print(f"MisclassificationCount={misclassification_count}")
48    print(f"MisclassificationRate={misclassification_rate}%")
49    print(f"Accuracy={accuracy}%")
50
51 import pandas as pd
52 df=pd.read_csv('/home/sahyadri/Desktop/CSV-Files/tennis.csv')
53 concept=str(list(df)[-1])
54 concept_list=set(df[concept])
55 Attr={}
56 for a in df.columns[:-1]:
57     Attr[a]=set(df[a])
58     print(f"{a}:{Attr[a]}")
59 conceptProbs,AttrConcept,probability_list=train(df,Attr,concept_list,concept)
60 examples=pd.read_csv('/home/sahyadri/Desktop/CSV-Files/tennis.csv')
61 test(examples.values,Attr,concept_list,conceptProbs,AttrConcept,probability_list)
```

OutLook:{'Overcast', 'Sunny', 'Rain'}

Jupyter Naive Bayesian Classifiers pg6 Last Checkpoint: 11/11/2021 (autosaved) 🐍 Logout

File Edit View Insert Cell Kernel Widgets Help   Trusted | Python 3 ○

🖫 ✛ ✂ 🗐 🗋 ↑ ↓ ▶ Run ■ C ⏭ Code ⌄ ⌨

```
55  Attr={}
56  for a in df.columns[:-1]:
57      Attr[a]=set(df[a])
58      print(f"{a}:{Attr[a]}")
59  conceptProbs,AttrConcept,probability_list=train(df,Attr,concept_list,concept)
60  examples=pd.read_csv('/home/sahyadri/Desktop/CSV-Files/tennis.csv')
61  test(examples.values,Attr,concept_list,conceptProbs,AttrConcept,probability_list)
```

```
OutLook:{'Overcast', 'Sunny', 'Rain'}
Temperature:{'Cool', 'Hot', 'Mild'}
Humidity:{'Normal', 'High'}
Wind:{'Weak', 'Strong'}
P(A):{'Yes': 0.6428571428571429, 'No': 0.35714285714285715}

P(X/A):{'OutLook': {'Overcast': {'Yes': 0.4444444444444444, 'No': 0.0}, 'Sunny': {'Yes': 0.2222222222222222, 'No':
0.6}, 'Rain': {'Yes': 0.3333333333333333, 'No': 0.4}}, 'Temperature': {'Cool': {'Yes': 0.3333333333333333, 'No': 0.
2}, 'Hot': {'Yes': 0.2222222222222222, 'No': 0.4}, 'Mild': {'Yes': 0.4444444444444444, 'No': 0.4}}, 'Humidity': {'No
rmal': {'Yes': 0.6666666666666666, 'No': 0.2}, 'High': {'Yes': 0.3333333333333333, 'No': 0.8}}, 'Wind': {'Weak': {'Y
es': 0.6666666666666666, 'No': 0.4}, 'Strong': {'Yes': 0.3333333333333333, 'No': 0.6}}}

P(X):{'OutLook': {'Overcast': 0.2857142857142857, 'Sunny': 0.35714285714285715, 'Rain': 0.35714285714285715}, 'Tempe
rature': {'Cool': 0.2857142857142857, 'Hot': 0.2857142857142857, 'Mild': 0.42857142857142855}, 'Humidity': {'Normal
': 0.5, 'High': 0.5}, 'Wind': {'Weak': 0.5714285714285714, 'Strong': 0.42857142857142855}}

{'Yes': 0.2419753086419753, 'No': 0.9408000000000002}
classfication:No Excepted:No
{'Yes': 0.16131687242798354, 'No': 1.8816000000000002}
classfication:No Excepted:No
```

In [ ]: 1

In [ ]: 1

localhost:8888/notebooks/Naive Bayesian Classifiers pg6.ipynb

110%

jupyter Naive Bayesian Classifiers pg6 Last Checkpoint: 11/11/2021 (autosaved)

Logout

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

Trusted   Python 3 O

Run   C   ⏭   Code   ⌨

```
55  Attr={}
56  for a in df.columns[:-1]:
57      Attr[a]=set(df[a])
58      print(f"{a}:{Attr[a]}")
59  conceptProbs,AttrConcept,probability_list=train(df,Attr,concept_list,concept)
60  examples=pd.read_csv('/home/sahyadri/Desktop/CSV-Files/tennis.csv')
61  test(examples.values,Attr,concept_list,conceptProbs,AttrConcept,probability_list)
```

```
classfication:No Excepted:No
{'Yes': 0.16131687242798354, 'No': 1.8816000000000002}
classfication:No Excepted:No
{'Yes': 0.6049382716049383, 'No': 0.0}
classfication:Yes Excepted:Yes
{'Yes': 0.4839506172839506, 'No': 0.4181333333333335}
classfication:Yes Excepted:Yes
{'Yes': 1.0888888888888888, 'No': 0.07840000000000004}
classfication:Yes Excepted:Yes
{'Yes': 0.7259259259259259, 'No': 0.15680000000000005}
classfication:Yes Excepted:No
{'Yes': 1.2098765432098766, 'No': 0.0}
classfication:Yes Excepted:Yes
{'Yes': 0.3226337448559671, 'No': 0.6272000000000001}
classfication:No Excepted:No
{'Yes': 0.7259259259259256, 'No': 0.11760000000000002}
classfication:Yes Excepted:Yes
{'Yes': 0.9679012345679012, 'No': 0.10453333333333338}
classfication:Yes Excepted:Yes
{'Yes': 0.43017832647462273, 'No': 0.31360000000000005}
```

In [ ]: 1

In [ ]: 1

```
55  Attr={}
56  for a in df.columns[:-1]:
57      Attr[a]=set(df[a])
58      print(f"{a}:{Attr[a]}")
59  conceptProbs,AttrConcept,probability_list=train(df,Attr,concept_list,concept)
60  examples=pd.read_csv('/home/sahyadri/Desktop/CSV-Files/tennis.csv')
61  test(examples.values,Attr,concept_list,conceptProbs,AttrConcept,probability_list)
```

```
{'Yes': 1.2098765432098766, 'No': 0.0}
classfication:Yes Excepted:Yes
{'Yes': 0.3226337448559671, 'No': 0.6272000000000001}
classfication:No Excepted:No
{'Yes': 0.7259259259259256, 'No': 0.11760000000000002}
classfication:Yes Excepted:Yes
{'Yes': 0.9679012345679012, 'No': 0.10453333333333338}
classfication:Yes Excepted:Yes
{'Yes': 0.43017832647462273, 'No': 0.31360000000000005}
classfication:Yes Excepted:Yes
{'Yes': 0.5377229080932785, 'No': 0.0}
classfication:Yes Excepted:Yes
{'Yes': 1.2098765432098766, 'No': 0.0}
classfication:Yes Excepted:Yes
{'Yes': 0.3226337448559671, 'No': 0.8362666666666669}
classfication:No Excepted:No
MisclassificationCount=1
MisclassificationRate=7.142857142857143%
Accuracy=92.85714285714286%
```

In [ ]: 1

In [ ]: 1